



TITLE:

Cross Assembler/Editor Program of HP2100 to Use INTEL 8085

AUTHOR(S):

Higo, Toshiyasu; Iwashita, Yoshihisa

CITATION:

Higo, Toshiyasu ...[et al]. Cross Assembler/Editor Program of HP2100 to Use INTEL 8085. Bulletin of the Institute for Chemical Research, Kyoto University 1980, 58(1): 1-4

ISSUE DATE:

1980-03-31

URL:

<http://hdl.handle.net/2433/76866>

RIGHT:

Cross Assembler/Editor Program of HP2100 to Use INTEL 8085

Toshiyasu HIGO and Yoshihisa IWASHITA*

Received November 25, 1979

To save the time of programming of a microprocessor INTEL 8085, a program of cross assembler with editor was developed. This program is executed on a minicomputer HP2100 and is written in FORTRAN II and ASSEMBLER. By the use of this program and a system of the HP2100, the time of programming is saved by an order of magnitude compared to the case of hand-assembling.

KEY WORDS: Computer program / Cross Assembler with Editor /
INTEL 8085 /

I. INTRODUCTION

Program development is a very much time-consuming work when one wants to use a microprocessor (μ P). However, if a large computer system is available, one can lessen the time of programming remarkably by providing a translation program of the computer system, fit to the specified μ P.¹⁾ This program (cross assembler) translates the symbolic source language instructions into object codes. We report here a Cross Assembler/Editor program of a minicomputer HP2100²⁾ developed in our laboratory to use the INTEL 8085.

II. FUNCTIONS OF CROSS ASSEMBLER/EDITOR

II-1. Principle

A primitive method to run a μ P is to load it with the machine codes. However this method needs a very tedious work of hand-assembling. This work can be omitted by the use of a minicomputer run with Cross Assembler/Editor program. The principle of operation is shown in Fig. 1.

The system arrangement to use a μ P is shown in Fig. 2. As shown in the figure, a source file is edited and stored in the main memory of a minicomputer through a TTY or a PTR, and the source file is assembled and output to a LP and/or a PTP. The object tape is read into the μ P off line. So that the time needed to run the μ P is saved by a large amount.

* 肥後寿泰, 岩下芳久: Laboratory of Nuclear Reaction, Institute for Chemical Research, Kyoto University.

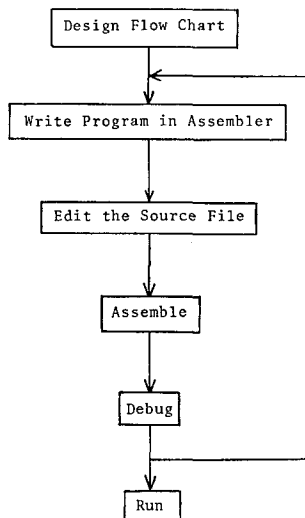


Fig. 1. Principle of operation to construct a program for a μ P.

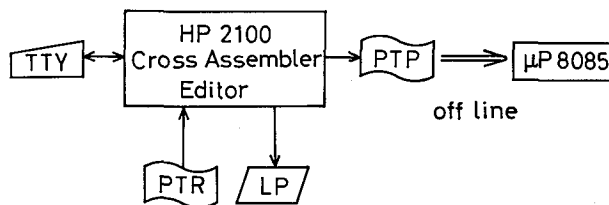


Fig. 2. System arrangement to use μ P 8085.

II-2. Functions of Editor Program

This program edits the source file which is stored in the memory of a minicomputer and has the following functions;

- 1) **Read** the source file or amendment informations from a TTY or a PTR and store them into the memory.
- 2) **List** a specified part or all of the source file.
- 3) **Delete, insert or replace** specified statements.
- 4) **Change** specified character string to another one.
- 5) **Search** specified character string.
- 6) **Output** the source file to a PTP.

Finally the program calls for the subprogram "ASMB" for assembling the source file.

II-3. Functions of Cross Assembler Program

This program is named "ASMB". This program checks mnemonic code grammatically and converts it into a machine code. In this program, the subprogram of code conversion and that of output for PTP are written in HP ASSEMBLER. Other subprograms are written in HP FORTRAN II.

Besides the instructions inherent in the μ P 8085, this program makes 6 pseudo instructions available. Those are as follows;

- 1) "ORG" — set the initial address to load,
- 2) "EQU" — define a symbol,
- 3) "SET" and 4) "DSET" — record a specified data into consecutive addresses,
- 5) "BSS" — reserve a block of memory,
- 6) "END".

The output tape is punched in any type of those three formats; 1. Binary format, 2. Intellex Hex format and 3. Hexadecimal format (Ascii code). Output in format 2 is used for the PROM WRITER of Data Processing Center, Kyoto University. Format 3 is the same as the format of output from the μ P system by the "Display" command in the monitor of the μ P 8085 (SDK-85),³⁾ and one can read the output in this format through a TTY.

III. PERFORMANCE

Figure 3 shows an example of the list output. When compared to the case that the HP symbolic editor are used to construct a source file, this editor program saves the time needed by 5 or 6 times. Cross Assembler program makes the time of assembling negligibly small.

1	1000			ORG	1000H
2	1234		DELAY	EQU	1234H
3	1000	31C820	ALN	LXI	SP,20C8H
4	1003	0607	LO	MVI	B,7
5	1005	211D10		LXI	H,DATA
6	1008	3E90	L1	MVI	A,90H
7	100A	7E	L2	MOV	A,M
8	100B	320018		STA	1800H
9	100E	CD3412		CALL	DELAY
10	1011	23		INX	H
11	1012	0D		DCR	C
12	1013	C20A10		JNZ	L2
13	1016	05		DCR	B
14	1017	C20810		JNZ	L1
15	101A	C30310		JMP	L0
16	101D		DATA	SET	5AH,9FH,4AH,0BH
		3A9F4A0B			
17	1021	2A99		DSET	992AH
18	1023			END	

DELAY	1234
ALN	1000
LO	1003
L1	1008
L2	100A
DATA	101D

NUMBER OF ERROR= 0

Fig. 3. An example of list output from LP by the Cross Assembler/Editor program. Column 1 shows the line number of source file. Column 4, 5, and 6 show an input source file; column 4-label, column 5-opcode and column 6-operand. Column 2 indicates addresses in hexadecimal format and column 3 shows output machine codes. After the list stated above is output, a table of labels are output and finally the number of grammatical errors in the source file is written.

Tedious work of debugging still remains. The time of debugging is saved very much if the μ P system run by the target program is simulated by the minicomputer.

A Cross Assembler/Editor program which includes this function is under development.

ACKNOWLEDGMENTS

The authors would like to acknowledge Professor H. Takekoshi and other staffs of the Laboratory of Nuclear Science, Institute for Chemical Research, for their encouragements in the course of developing the program. They are also grateful to Professor T. Yanabu for his critical reading of this manuscript.

REFERENCES

- (1) P. J. Horne, *IEEE Transactions on Nuclear Science*, **NS-24**, 1772 (1977).
- (2) Hewlett-Packard company, "A Pocket Guide to the 2100 Computer", (Hewlett-Packard co., Cupertino, California, 1972).
- (3) Intel corporation, "SDK-85 System Design Kit User's Manual", (Intel corp., Santa Clara, California, 1977).